IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **DeWitt, Jr. et al.**	§	Confirmation No.: **4157**
Serial No. **10/757,186**	§	Group Art Unit: **2111**
Filed: **January 14, 2004**	§	Examiner: **Dang, Khanh**
For: **Method and Apparatus for Qualifying collection of Performance Monitoring Events by Types of Interrupt When Interrupt Occurs**	§	

35525

PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on February 29, 2008.

A fee of $510.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees, which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

This appeal has no related proceedings or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

The claims in the application are: 1-23

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims canceled: None

Claims withdrawn from consideration but not canceled: None

Claims pending: 1-23

Claims allowed: None

Claims rejected: 1-23

Claims objected to: None

C. CLAIMS ON APPEAL

The claims on appeal are: 1-23

STATUS OF AMENDMENTS

An Amendment after the Final Office Action dated November 29, 2007, was not filed. Accordingly, the claims on appeal herein are as presented in the Response to Office Action dated November 1, 2007.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to a data processing system for qualifying events when an interrupt occurs. The data processing system comprises an interrupt unit control register for indicating an interrupt type selected to be monitored (page 21, lines 17-29; **Figure 3**, **308**) and an interrupt unit (**Figure 3**, **304**), responsive to an interrupt occurring during code execution, for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register (page 21, lines 17- page 22, line 2). The data processing system also comprises a performance monitoring unit (**Figure 3**, **306**) and one or more hardware counters (**Figure 3**, **312**, **314**) located within the performance monitoring unit; wherein the one or more hardware counters count events that occur during processing of the interrupt responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register (page 21, lines 17- page 22, line 2).

B. CLAIM 8 - INDEPENDENT

The subject matter of claim 8 is directed to a method of executing instructions in a data processing system. A signal is received at a microprocessor of the data processing system for invoking an interrupt, wherein the invoked interrupt includes a plurality of states (page 23, line 26- page 24, line 5, page 25, line 2-6, page 27, line 2; **Figure 4**, **416**). At least one event for a selected state of the plurality of states is counted during processing of the invoked interrupt (page 24, lines 6-15; **Figure 4**, **426**).

C. CLAIM 16 - INDEPENDENT

The subject matter of claim 16 is directed to a computer program product stored in a computer readable medium in a data processing system for processing instructions. The computer program product comprises first instructions for receiving a signal at a microprocessor of the data processing system for invoking an interrupt, wherein the invoked interrupt includes a plurality of

states (page 23, line 26- page 24, line 5, page 25, line 2-6, page 27, line 2; **Figure 4**, **416**). Second instructions for counting at least one event for a selected state of the plurality of states during processing of the invoked interrupt (page 24, lines 6-15; **Figure 4**, **426**).

D. CLAIM 5 - DEPENDENT

The subject matter of claim 5 is directed to the system of claim 1, wherein the one or more hardware counters count events that occur during processing of the interrupt based on the interrupt type of the interrupt (page 24, line 22 – page 25, line 6, page 25, line 28 – page 26, line 30; **Figure 5**, **512-532**).

E. CLAIM 7 - DEPENDENT

The subject matter of claim 7 is directed to the system of claim 1, wherein the interrupt is a first interrupt, and further comprising a second interrupt that interrupts the first interrupt, wherein the one or more hardware counters separately count the events that occur during the processing of the first interrupt and events that occur during processing of the second interrupt (page 20, lines 24-page 21, line 5).

F. CLAIM 15 - DEPENDENT

The subject matter of claim 15 is directed to the method of claim 8, wherein the interrupt is a first interrupt, and further comprising a second interrupt that interrupts the first interrupt, wherein the one or more hardware counters separately count the events that occur during the processing of the first interrupt and events that occur during processing of the second interrupt (page 20, lines 24-page 21, line 5).

G. CLAIM 22 - DEPENDENT

The subject matter of claim 22 is directed to the computer program product of claim 16, wherein the interrupt is a first interrupt, and further comprising a second interrupt that interrupts the first interrupt, wherein the one or more hardware counters separately count the events that occur during the processing of the first interrupt and events that occur during processing of the second interrupt (page 20, lines 24-page 21, line 5).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal are as follows:

A. GROUND OF REJECTION 1

Whether claims 1-7 satisfy the requirements of 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which Appellants regard as the invention

B. GROUND OF REJECTION 2

Whether claims 1-7 satisfy the requirements of 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement.

C. GROUND OF REJECTION 3

Whether claims 1-6, 8-12, and 16-21 are anticipated by *Levine*, under 35 U.S.C. § 102(b)

D. GROUND OF REJECTION 4

Whether the Examiner failed to state a *prima facie* obviousness rejection under 35 U.S.C. § 103 against claims 1-6, 8-14, and 16-21 in view of the admitted prior art in view of *Levine et al.*, U.S. Patent No. 5,691,920 (hereinafter "*Levine*").

E. GROUND OF REJECTION 5

Whether the Examiner failed to state a *prima facie* obviousness rejection under 35 U.S.C. § 103 against claims 7, 15, and 22 in view of *Levine* in view of "Computer System Architecture" by *Morris Mano* (hereinafter "*Mano*")or as being unpatentable over the admitted prior art in view of *Levine* and *Mano*.

There is also a 35 U.S.C. § 101 rejection against claims 16 and 23. Appellants are not appealing this rejection. Appellants agree to amend the claims as necessary at the appropriate time.

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 1-7)

Claims 1-7 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which Appellants regard as the invention.

In rejecting the claims, the Examiner states:

> Claims 1-7 are directed to an apparatus. However, the essential structural cooperative relationship(s) between the so-called "interrupt unit control mechanism," "interrupt unit," and "performance monitoring unit" have been omitted, such omission amounting to a gap between the necessary structural connections. See MPEP § 2172.01.
>
> MPEP 2172.01 requires that relationships between elements recited in the claims must be specified. Specifically, MPEP 2172.02 requires interrelation and structural relationships between essential elements in the claims. Therefore, it is the Examiner's position that the claimed elements, as defined in the originally filed specification and as identified above, are essential elements to the claimed invention. Since they are essential elements as defined in the originally filed specification, their structural cooperative relationships must be provided in the claims. Further, it is also the Examiner's position that the claimed elements, as identified above, function simultaneously, are directly functionally related, directly inter-cooperate, and/or serve independent purposes, as evidenced from the originally filed specification.
>
> If Applicants disagree with the Examiner that the above identified elements, as defined by the originally filed specification, are essential elements to the claimed invention, and that the above identified elements are directly functionally related, directly inter-cooperate, and/or serve independent purposes, it is requested that Applicants provide evidences showing that the identified elements are not essential elements to the claimed invention, do not function simultaneously, are not directly functionally related, do not directly inter-cooperate, and/or do not serve independent purposes; and state on the record that this is the case.

Final Office Action dated November 29, 2007, pp. 2-3.

Further, on page 14 of the Final Office Action, mailed November 29, 2007, the Examiner suggests that the word "coupled" be provided to overcome the rejection. However, Appellants respectfully disagree that that the term coupled needs to be added to the claims.

Further, Appellants respectfully submit that claims 1-7 are definite and the inter-relationship of the components are set forth in the last element of claim 1, which recites "one or

more hardware counters located within the performance monitoring unit; wherein the one or more hardware counters count the occurrence of events that occur during processing of the interrupt responsive to a determination by the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register of the selected type." Thus, the hardware counters are located within the performance monitoring unit and the hardware counters count the occurrence events in response to the interrupt unit determining that the interrupt is of an interrupt type that has been selected to be monitored. The interrupt unit determines this by checking the interrupt unit control registers, which indicate the interrupt type that is to be monitored. The specification fully supports this interpretation on page 21, line 17 – page 22, line 2, and page 25, line 28 – page 27, line 6; **Figure 5.**

Claims 2-7 are rejected solely on the basis of depending from claim 1.

Appellants respectfully submit that claims 1-7 as presented herein fully satisfy the requirements of 35 U.S.C. § 112, second paragraph, and that the rejection of the claims as being indefinite should be withdrawn.

B. GROUND OF REJECTION 2 (Claims 1-7)

Claims 1-7 stand rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement.

In rejecting the claims, the Examiner states:

> The claims contain subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Specifically, the language "responsive ... control register" (lines 11-12) lack clear support from the specification. See also the 35 USC 112, 2nd paragraph above.

Final Office Action dated November 29, 2007, p. 3.

Appellants respectfully disagree with the Examiner's position. Appellants submit that the specification, on page 21, line 17 – page 22, line 2, fully supports the recited feature of "responsive to a determination by the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register." The specification, on page 21, line 26-28, states "IUCR such as IUCR1 **308** includes a type field that indicates which interrupt type is to generate a performance monitoring counter signal." Thus, the specification clearly

supports s that the interrupt unit control register indicates the interrupt type selected to be monitored. Further, on page 21, line 28 – page 22, line 2, the specification states "The type field in the IUCR is later examined by interrupt unit **304** to see if it is an interrupt type of interest, *i.e.*, whether events are to be counted during execution of the interrupt." Thus, the specification clearly supports that the interrupt unit determines that the interrupt is of the interrupt type to be monitored as indicated by the interrupt unit control register, as the specification states that interrupt unit **304** examines the interrupt unit control register to see if the interrupt is an interrupt type of interest.

Further **Figure 5** and the text describing **Figure 5** on page 25, line 28 – page 27, line 6, also describe this process in detail. Specifically, the specification states, "When an interrupt occurs during code execution (step **512**), the interrupt unit examines the IUCR type field (step **514**). A determination is made as to whether the interrupt is the same type as the IUCR type, which is an interrupt type of interest (step **516**)." Thus, the specification clearly supports the claimed feature of "responsive to a determination by the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register."

Thus, Appellants submit that the specification clearly supports the feature of "responsive to a determination by the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register." Further, claims 2-7 were rejected on the basis of depending from claim 1.

Appellants respectfully submit that the claim 1-7, as currently drafted fully satisfy the requirements of 35 U.S.C. § 112, first paragraph, in all respects, and that the rejection of claims 1-7 under 35 U.S.C. § 112, first paragraph, should be withdrawn.

C. GROUND OF REJECTION 3 (Claims 1-6, 8-12, and 16-21)

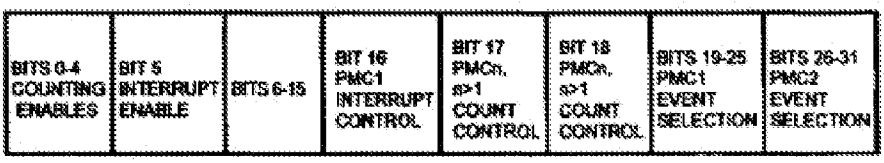
Claims 1-6, 8-12, and 16-21 stand rejected under 35 U.S.C. § 102(b) as being anticipated by *Levine et al.*, U.S. Patent No. 5,691,920 (hereinafter "*Levine*").

C.1. Claims 1-6

In rejecting claim 1, the Examiner states:

> Referring to claim 1: Levine discloses a performance-monitoring unit (Abstract) and one or more hardware counters (column 10, lines 58-59) located within the performance-monitoring unit (figure 4, structures 50 and 51). Levine

discloses a variety of operations of the performance-monitoring unit (column 10, lines 34-56); Levine discloses setting the performance-monitoring unit to selectively monitor the instructions within specific addresses (column 12, 2nd paragraph). The instructions within specific addresses are a selected interrupt handling routine, and the Levine's selective monitoring on the particular instructions is the claimed hardware counters' counting the occurrence of events during an interrupt of a selected type. Furthermore, Levine disclosed two separate counters for selected events (figure6A, column 8, lines 57-60). It is also clear that in Levine, at least bits 5 and 16 in a register field of a particular MMCR associated with a particular counter PMC indicate "an interrupt of a selected type" or specifically, the performance monitoring interrupt type. In other words, the type of interrupt indicated and associated with MMCR shown below is the performance monitoring interrupt type.



MONITOR MODE CONTROL REGISTER 0
(MMCR0)

FIGURE 6A

Further, it is also clear the performance monitoring interrupt is presented to the interrupt resolution logic or interrupt handler 57, which uses a plurality of interrupt handling routines to serve the interrupts depending from the types of interrupts. In other words, depending on a particular type of interrupt, the interrupt handler 57 will select a particular interrupt handling routine among the plurality of interrupt handling routines employed by the interrupt handler 57. It is clear that the interrupt handler 57 MUST be able to determine which type of interrupt presented to it before it selects the corresponding interrupt handling routine accordingly. In the instant case, the interrupt handler 57 must be able to determine that the presented interrupt is the performance monitoring interrupt before it selects a routine designed for performance monitoring. Thus, it is clear that the interrupt resolution logic or interrupt handler 57 is readable as the "interrupt unit."

Final Office Action dated November 29, 2007, pp. 4-6.

Claim 1 recites:

1. A data processing system for qualifying events when an interrupt occurs, comprising:

an interrupt unit control register for indicating an interrupt type selected to be monitored;

an interrupt unit, responsive to an interrupt occurring during code execution, for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register;

a performance monitoring unit; and

one or more hardware counters located within the performance monitoring unit; wherein the one or more hardware counters count events that occur during processing of the interrupt responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.,* 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case, each and every feature of the presently claimed invention is not identically shown in the cited reference, arranged as they are in the claims. With respect to claim 1, in particular, *Levine* does not disclose or suggest "an interrupt unit control register for indicating an interrupt type selected to be monitored", "an interrupt unit, responsive to an interrupt occurring during code execution, for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register", or "one or more hardware counters located within the performance monitoring unit; wherein the one or more hardware counters count events that occur during processing of the interrupt responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register."

Initially, *Levine* does not disclose or in any way suggest, "an interrupt unit control register for indicating an interrupt type selected to be monitored" as presently recited in claim 1. *Levine* does not describe interrupt types and does not disclose a register that indicates an interrupt type selected to be monitored. Bits 5 and 16 illustrated in Figure 6A of *Levine* reproduced by the Examiner are not interrupt types selected to be monitored, but, at best, may relate to states of an interrupt.

In the Final Office Action mailed November 29, 2007, the Examiner states, on page 16 that:

> at least bits 5 and 16 in a register field of a particular MMCR associated with a particular counter PMC indicate "an interrupt of a selected type" or specifically, the performance monitoring interrupt type. In other words, the type of interrupt indicated and associated with MMCR shown below is the performance monitoring interrupt type.

Appellants respectfully disagree with the Examiner's conclusion. *Levine* discusses Figure 6A in the passage from column 10, line 57 – column 11, line 54, which is reproduced below:

> In FIG. 6a, an example representation of one configuration of MMCR0 suitable for controlling the operation of two PMC counters, e.g., PMC1 and PMC2, is illustrated. As shown in the example, MMCR0 is partitioned into a number of bit fields whose settings select events to be counted, enable performance monitor interrupts, specify the conditions under which counting is enabled, and set a threshold value (X).
>
> The threshold value (X) is both variable and software selectable and its purpose is to allow characterization of certain data, such that by accumulating counts of accesses that exceed decreasing threshold values, designers gain a clearer picture of conflicts. The threshold value (X) is considered exceeded when a decrementer reaches zero before the data instruction completes. Conversely, the threshold value is not considered exceeded if the data instruction completes before the decrementer reaches zero. Of course, depending on the data instruction being executed, completed has different meanings. For example, for a load instruction, "completed" indicates that the data associated with the instruction was received, while for a "store" instruction, "completed" indicates that the data was successfully written. A user readable counter, e.g., PMC1, suitably increments every time the threshold value is exceeded.
>
> A user may determine the number of times the threshold value is exceeded prior to the signalling of performance monitor interrupt. For example, the user may set initial values for the counters to cause an interrupt on the 100th data miss that exceeds the specified threshold. With the appropriate values, the PM facility is readily suitable for use in identifying system performance problems.
>
> Referring to FIG. 6a, as illustrated by this example, bits 0-4 and 18 of the MMCR0 determine the scenarios under which counting is enabled. By way of example, in a preferred embodiment, bit 0 is a freeze counting bit (FC). When at a high logic level (FC=1), the values in PMCn counters are not changed by hardware events, i.e., counting is frozen. When bit 0 is at a low logic level (FC=0), the values of the PMCn can be changed by chosen hardware events. Bits 1-4 indicate other specific conditions under which counting is frozen.
>
> For example, bit 1 is a freeze counting while in a supervisor state (FCS) bit, bit 2 is a freeze counting while in a problem state (FCP) bit, bit 3 is a freeze counting while PM=1 (FCPM1) bit, and bit 4 is a freeze counting while PM=0

(FCPM0) bit. PM represents the performance monitor marked bit, bit 29, of a machine state register (MSR) (an SPR 40, FIG. 1). For bits 1 and 2, a supervisor or problem state is indicated by the logic level of the PR (privilege) bit of the MSR. The states for freezing counting with these bits are as follows: for bit 1, FCS=1 and PR=0; for bit 2, FCP=1 and PR=1; for bit 3, FCPM1=1 and PM=1; and for bit 4, FCPM0=1 and PM=0. The state for allowing counting with these bits are as follows: for bit 1, FCS=1 and PR=1; for bit 2, FCP=1 and PR=0; for bit 3, FCPM1=1 and PM=0; and for bit 4, FCPM0=1 and PM=1.

Bits 5, 16, and 17 are utilized to control interrupt signals triggered by PMCn. Bits 6-9 are utilized to control the time or event-based transitions. The threshold value (X) is variably set by bits 10-15. Bit 18 control counting enablement for PMCn, n>1, such that when low, counting is enabled, but when high, counting is disabled until bit 0 of PMC1 is high or a performance monitoring exception is signaled. Bits 19-25 are used for event selection, i.e, selection of signals to be counted, for PMC1.

As can be seen, *Levine* clearly teaches generating a monitor interrupt. Specifically, in column 11, lines 13-19, *Levine* teaches that a threshold value is set by a user, then when that value is exceeded, provided that bit 5 is set to enable interrupts, a monitor interrupt is generated. Thus, *Levine* teaches a register for generating interrupt signals based on a threshold. In contradistinction, claims 1 recites, "an interrupt unit control register for indicating an interrupt type selected to be monitored." *Levine* does not teach selecting a type of interrupt to be monitored, monitoring interrupts, or an interrupt unit control register that indicates an interrupt type selected to be monitored. *Levine* is silent in regards to "an interrupt unit control register for indicating an interrupt type selected to be monitored." Thus, *Levine* fails to teach or suggest the feature of "an interrupt unit control register for indicating an interrupt type selected to be monitored."

Furthermore, *Levine* does not disclose "an interrupt unit, responsive to an interrupt occurring during code execution, for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register" as recited in claim 1. In rejecting the claims, the Examiner construes the interrupt handler 57 in *Levine* as corresponding to the interrupt unit recited in claim 1. In particular, the Examiner points out that interrupt handler 57 in *Levine* selects one of interrupt handling routines 71, 77 and 79 illustrated in Figure 4, and concludes "[i]t is clear that the interrupt handler 57 MUST be able to determine which type of interrupt presented to it before it selects the corresponding interrupt handling routine accordingly." Appellants respectfully disagree.

Appellants respectfully submit that *Levine* nowhere discloses or suggests that interrupt handler 57 selects any of interrupt handling routines 71, 77 and 79 based on interrupt type, and *Levine* certainly does not disclose that interrupt handler 57 selects a particular interrupt handling routine based on an interrupt type selected to be monitored. Neither the Monitor Mode Control Register illustrated in Figure 6A referred to by the Examiner nor the discussion of the interrupt handling routines in column 9, lines 46-62 in *Levine* suggests that interrupt handler 57 selects a particular interrupt routine based on interrupt type. Rather, *Levine* merely teaches, in column 9, lines 46-62, that interrupt resolution logic passes the interrupt to one of various interrupt handler routines. However, passing an interrupt to one of multiple interrupt handling routines has nothing to do with determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register. Even assuming *arguendo* that the interrupt handler selects a specific interrupt handler routine based on a type of interrupt, selecting an appropriate interrupt handler routing is not the same as, and has nothing to do with "determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register." Levine does not disclose that interrupt handler 57 functions to be "responsive to an interrupt occurring during code execution for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register" as recited in claim 1.

Levine also does not disclose or suggest "one or more hardware counters located within the performance monitoring unit; wherein the one or more hardware counters count events that occur during processing of the interrupt responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register." Although *Levine* may disclose counters in performance monitor 50 illustrated in Figure 4 of *Levine*, the counters are described as being for counting "processor/storage related events" (see col. 8, lines 33-41 of Levine). *Levine* does not disclose that the counters count events that occur during processing of an interrupt "responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register" as required in claim 1. *Levine* does not disclose interrupt types and does not determine if an interrupt that occurs during code execution is of an interrupt type selected to be monitored.

For at least all the above reasons, *Levine* does not identically show every element recited in claim 1 arranged as they are in the claim, and does not anticipate claim 1.

Claims 2-6 depend from and further restrict claim 1, and are also not anticipated by *Levine*, at least by virtue of their dependency. Furthermore, many of these claims recite additional features that are not disclosed or suggested by *Levine*.

Claim 5 recites the feature of "wherein the one or more hardware counters count events that occur during processing of the interrupt based on the interrupt type of the interrupt." Since, as indicated above, *Levine* does not disclose different interrupt types as claimed, *Levine* also does not disclose counting events during an interrupt according to the interrupt type of the interrupt. Claim 5, accordingly, patentably distinguishes over *Levine* in it own right as well as by virtue of its dependency.

C.2. Claims 8-12 and 16-21

Independent claim 8 recites as follows:

> 8. A method of executing instructions in a data processing system, comprising:
>
> receiving a signal at a microprocessor of the data processing system for invoking an interrupt, wherein the invoked interrupt includes a plurality of states; and
>
> counting at least one event for a selected state of the plurality of states during processing of the invoked interrupt.

In rejecting claim 8, the Examiner states:

> Referring to claims 8-10: It is clear that the states of the performance monitor interrupt include at least the state wherein an interrupt signal is generated, and a state wherein the interrupt is serviced or processed. As discussed above, during servicing or processing the performance monitor interrupt, multiple events are counted by hardware counters located inside the performance monitor unit 50. The counters count the occurrence of events during the interrupt service routine, which is a "state" of the performance monitor interrupt.

Final Office Action dated November 29, 2007, pp. 6-7.

Claim 8 recites, "the invoked interrupt includes a plurality of states." Further, claim 8 recites the feature of "counting at least one event for a selected state of the plurality of states during processing of the invoked interrupt." *Levine* does not disclose

an invoked interrupt that includes a plurality of states, and also does not disclose counting at least one event for a selected state of the plurality of states during processing of the invoked interrupt. Rather, in column 9, lines 55-62, *Levine* teaches that when an interrupt is generated (signaled), the state of various execution units are saved in a saved state register (SSR). Further, when the interrupt is actually serviced, the content of the SSR provides information concerning instructions **at the time of signaling the interrupt**. In contradistinction, claim 8 recites, "counting at least one event for a selected state of the plurality of states during processing of the invoked interrupt." Thus, ***Levine* teaches providing state information of the system at the time the interrupt was generated**; whereas claim 8 recites, "counting at least one event for a selected state of the plurality of states during processing of the invoked interrupt." Thus, *Levine* fails to teach the feature of "counting at least one event for a selected state of the plurality of states during processing of the invoked interrupt."

Furthermore, nothing in *Levine* teaches, "wherein the invoked interrupt includes a plurality of states." Rather, as taught by *Levine*, when an interrupt is generated, state information of various execution units is saved to SSRs. Nothing in *Levine* teaches that the generated interrupt "includes a plurality of states." Therefore, *Levine* fails to teach the feature of "receiving a signal at a microprocessor of the data processing system for invoking an interrupt, wherein the invoked interrupt includes a plurality of states."

Therefore, for at least the reasons set forth above, Appellants submit that *Levine* fails to teach all the features of claim 8. Claim 8, accordingly, is not anticipated by *Levine* and patentably distinguishes over *Levine* in its present form.

Independent claim 16 recites similar subject matter as claim 8, and is also not anticipated by *Levine* for similar reasons as claim 8. Claims 9-12 and 17-21 depend from and further restrict one of claims 8 and 16 and are also not anticipated by *Levine*.

Therefore, Appellants respectfully request that the rejection of claims 1-6, 8-12, and 16-21 under 35 U.S.C. § 102(b) be withdrawn.

D. GROUND OF REJECTION 4 (Claims 1-6, 8-14, and 16-21)

Claims 1-6, 8-14, and 16-21 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the admitted prior art in view of *Levine*.

In rejecting the claims, the Examiner states:

> Referring to claim 1: The admitted prior art discloses a performance-monitoring unit (Specification, page 3, last paragraph, line 6) and one or more hardware counters (Specification, page 3, last paragraph, lines 1-2) located within the performance-monitoring unit. The admitted prior art does not disclose that the one or more hardware counters count the occurrence of events during an interrupt of a selected type.
>
> Levine discloses a variety of operations of the performance-monitoring unit (column 10, lines 34-39); Levine discloses setting the performance-monitoring unit to selectively monitor the instructions within specific addresses (column 12, 2"d paragraph). The instructions within specific addresses are a selected interrupt handling routine, and the Levine's selective monitoring on the particular instructions is the claimed hardware counters' counting the occurrence of events during an interrupt of a selected type. Levine teaches one to analyze the system performance and to focus particular sets of instructions for examining the performance bottleneck. Hence, it would have been obvious to one having ordinary skill in the computer art at the time Applicant made the invention to adapt Levine's teaching onto the admitted prior art because Levine teaches one to analyze the system performance and to focus particular sets of instructions for examining the performance bottleneck.

Final Office Action dated November 29, 2007, p. 9.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. §103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). Additionally, all limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Therefore, no *prima facie* obviousness rejection can be established if the proposed combination does not teach all of the features of the claimed invention. Furthermore, if an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988).

In the present case, neither the admitted prior art nor *Levine* nor their combination discloses or suggests "an interrupt unit control register for indicating an interrupt type selected to be monitored", "an interrupt unit, responsive to an interrupt occurring during code execution, for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by

the interrupt unit control register", or "one or more hardware counters located within the performance monitoring unit; wherein the one or more hardware counters count events that occur during processing of the interrupt responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register" as recited in claim 1. *Levine* does not disclose the subject matter of claim 1 for reasons discussed in detail above in Section C.

The admitted prior art is cited as only disclosing a performance monitoring unit having hardware counters. The admitted prior art does not teach the features of "an interrupt unit control register for indicating an interrupt type selected to be monitored", "an interrupt unit, responsive to an interrupt occurring during code execution, for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register", or "one or more hardware counters located within the performance monitoring unit; wherein the one or more hardware counters count events that occur during processing of the interrupt responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register." Thus the admitted prior art fails to teach or suggest these features. Accordingly, neither the admitted prior art, Levine, nor the combination of the admitted prior art in view of Levine teaches all of the features of the claimed invention as recited in claim 1. Thus, the Examiner has not established a *prima facie* case of obviousness in rejecting claim 1.

For similar reasons as discussed above, the admitted prior art in view of *Levine* also does not teach all of the features of the claimed invention as recited in claims 8 and 16. Thus, the Examiner has not established a *prima facie* case of obviousness in rejecting claims 8 and 16 as well.

Claims 1-6, 8-14, and 16-21 are; therefore, not obvious in view of the admitted prior art and *Levine*, and are allowable thereover.

Therefore, Appellants respectfully submit that the rejection of claims 1-6, 8-14, and 16-21 under 35 U.S.C. § 103(a) be withdrawn.

E. GROUND OF REJECTION 5 (Claims 7, 15, and 22)

Claims 7, 15, and 22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Levine in view of "Computer System Architecture" by Morris Mano or as being unpatentable over the admitted prior art in view of Levine and Mano.

In rejecting the claims, the Examiner states:

> Referring to claims 7, 15, and 22: The disclosures of the admitted prior art and Levine are stated above. As stated above, Levine discloses monitoring the particular interrupt according to the instructions address, and Levine discloses two separate counters for event selections (figure 6A, column 8, lines 57-60); thus, Levine discloses the hardware counters counting events separately. But neither explicitly discloses a second interrupt interrupts a first interrupt.
>
> Mano, as a popular academic textbook, discloses managing interrupt according to its priority (pages 434-435). Mano discloses that a higher priority interrupt can interrupt an in-process lower priority interrupt (page 435, 2nd paragraph). Mano teaches one to manage the limited system resources by prioritizing interrupt. Hence, it would have been obvious to one having ordinary skill in the computer art to adapt Mano's teaching onto the admitted prior art and Levine because Mano teaches one to manage the limited system resources by prioritizing interrupt.

Final Office Action dated November 29, 2007, page 13.

Claims 7, 15, and 22 depend from and further restrict one of independent claims 1, 8, and 16. *Mano* does not cure the deficiencies in *Levine* or in the admitted prior art in view of *Levine* as discussed above. *Mano* discloses managing interrupts according to a priority for the interrupt. *Mano* is silent in regards to the features of "an interrupt unit control register for indicating an interrupt type selected to be monitored", "an interrupt unit, responsive to an interrupt occurring during code execution, for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register", or "one or more hardware counters located within the performance monitoring unit; wherein the one or more hardware counters count events that occur during processing of the interrupt responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register." Thus *Mano* fails to teach or suggest the feature of "an interrupt unit control register for indicating an interrupt type selected to be monitored", "an interrupt unit, responsive to an interrupt occurring during code execution, for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control

register", or "one or more hardware counters located within the performance monitoring unit; wherein the one or more hardware counters count events that occur during processing of the interrupt responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register," as recited in claim 1.

Therefore, neither the combination of *Levine* in view of *Mano* nor the combination of the admitted prior art in view of *Levine* and further in view of *Mano* teaches or suggests the features of claims 1, 8, and 16. Thus, Appellants submit that claims 1, 8 and 16 are in condition for allowance over the cited combinations of *Levine* in view of *Mano* and the admitted prior art in view of *Levine* and further in view of *Mano*. Claims 7, 15 and 22, accordingly, are allowable in their present form at least by virtue of their dependency.

Therefore, Appellants respectfully submit that the rejection of claims 7, 15, and 22 under 35 U.S.C. § 103(a) be withdrawn.

F. CONCLUSION

As shown above, the Examiner has failed to state valid rejections against any of the claims. Therefore, Appellants request that the Board of Patent Appeals and Interferences reverse the rejections.

/Gerald H. Glanzman/

Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777

GG/blj

CLAIMS APPENDIX

The text of the claims involved in the appeal is as follows:

1. A data processing system for qualifying events when an interrupt occurs, comprising:

an interrupt unit control register for indicating an interrupt type selected to be monitored;

an interrupt unit, responsive to an interrupt occurring during code execution, for determining whether the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register;

a performance monitoring unit; and

one or more hardware counters located within the performance monitoring unit; wherein the one or more hardware counters count events that occur during processing of the interrupt responsive to the interrupt unit determining that the interrupt is of the interrupt type selected to be monitored as indicated by the interrupt unit control register.

2. The system of claim 1, wherein the one or more hardware counters count events that occur during a state of the interrupt.

3. The system of claim 2, wherein the state of the interrupt includes one of interrupt on, interrupt taken and interrupt acknowledged.

4. The system of claim 1, wherein the one or more hardware counters count multiple types of events that occur during the processing of the interrupt.

5. The system of claim 1, wherein the one or more hardware counters count events that occur during processing of the interrupt based on the interrupt type of the interrupt.

6. The system of claim 1, wherein the events include clock cycles and cache misses.

7. The system of claim 1, wherein the interrupt is a first interrupt, and further comprising a second interrupt that interrupts the first interrupt, wherein the one or more hardware counters separately count the events that occur during the processing of the first interrupt and events that occur during processing of the second interrupt.

8. A method of executing instructions in a data processing system, comprising:

receiving a signal at a microprocessor of the data processing system for invoking an interrupt, wherein the interrupt includes a plurality of states; and

counting at least one event for a selected state of the plurality of states during processing of the interrupt.

9. The method of claim 8, wherein the step of counting includes counting at least one event for each of the plurality of states during the processing of the interrupt.

10. The method of claim 8, wherein the plurality of states include interrupt on, interrupt taken and interrupt acknowledged.

11. The method of claim 8, wherein the at least one event includes clock cycles and cache misses.

12. The method of claim 8, wherein the step of counting includes counting multiple types of events for the selected state during the processing of the interrupt.

13. The method of claim 8, wherein the step of counting is performed by one or more hardware counters during the processing of the interrupt.

14. The method of claim 8, wherein the at least one event is counted based on an interrupt type of the interrupt during which the at least one event occurs.

15. The method of claim 8, wherein the interrupt is a first interrupt, and further comprising a second interrupt that interrupts the first interrupt, and wherein hardware counters separately count the at least one event that occurs during the processing of the first interrupt and at least one event that occurs during processing of the second interrupt.

16. A computer program product stored in a computer readable medium in a data processing system for processing instructions, the computer program product comprising:

first instructions for receiving a signal at a microprocessor of the data processing system for invoking an interrupt, wherein the interrupt includes a plurality of states; and

second instructions for counting at least one event for a selected state of the plurality of states during processing of the interrupt.

17. The computer program product of claim 16, wherein the plurality of states include interrupt on, interrupt taken and interrupt acknowledged.

18. The computer program product of claim 16, wherein the at least one event includes clock cycles and cache misses.

19. The computer program product of claim 16, wherein the second instructions comprise instructions for counting count multiple types of events for the selected state during the processing of the interrupt.

20. The computer program product of claim 16, wherein the counting is performed by one or more hardware counters during the processing of the interrupt.

21. The computer program product of claim 16, wherein the at least one event is counted based on an interrupt type of the interrupt during which the at least one event occurs.

22. The computer program product of claim 16, wherein the interrupt is a first interrupt, and further comprising a second interrupt that interrupts the first interrupt, and wherein hardware counters separately count the at least one event that occurs during the processing of the first interrupt and at least one event that occurs during processing of the second interrupt.

23. A computer implemented method for executing instructions stored in a computer readable medium for qualifying events when an interrupt occurs, comprising:

indicating a selected interrupt type to be monitored;

determining, responsive to an interrupt occurring during code execution, whether the interrupt is of the interrupt type selected to be monitored; and

responsive to determining that the interrupt is of the interrupt type selected to be monitored, counting, by one or more hardware counters located within a performance monitoring unit, events occurring during processing of the interrupt.

EVIDENCE APPENDIX

This appeal brief presents no additional evidence.

RELATED PROCEEDINGS APPENDIX

This appeal has no related proceedings.